

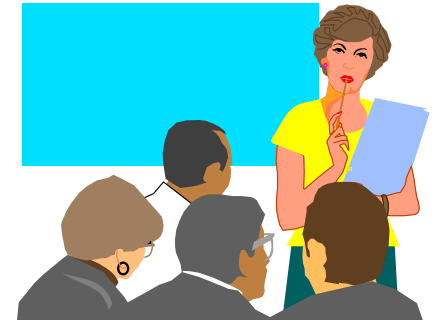
Week 3

- Stepwise Refinement
- Decisions with if - else
- Conditional Operator
- Switch statement
- Quiz

Read Pages 68 - 88

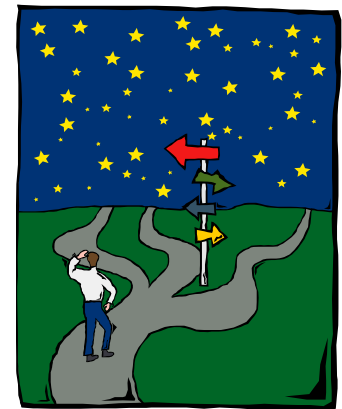
Initial Specification

- Write a program to manage withdrawals from a bank account.
- If there are enough funds, the program should accept the withdrawal, otherwise it should reject it.



Analyze Requirements ...

- What is the initial balance?
- When can a customer withdraw ?
- Are there any withdrawal fee ?
- What kind of outputs should the program produce?



Refining the Specifications

- The program should accept the initial balance, and then accept a withdrawal.
- If balance more than withdrawal amount, program should accept withdrawal, otherwise it should reject it. No withdrawal fee.
- The program should also display the new balance at the end of transaction.
- Input comes from the keyboard and output goes to the screen.

Enter initial balance: 150

Enter withdrawal: 50

Withdrawal accepted.

New balance: 100

Enter initial balance: 50

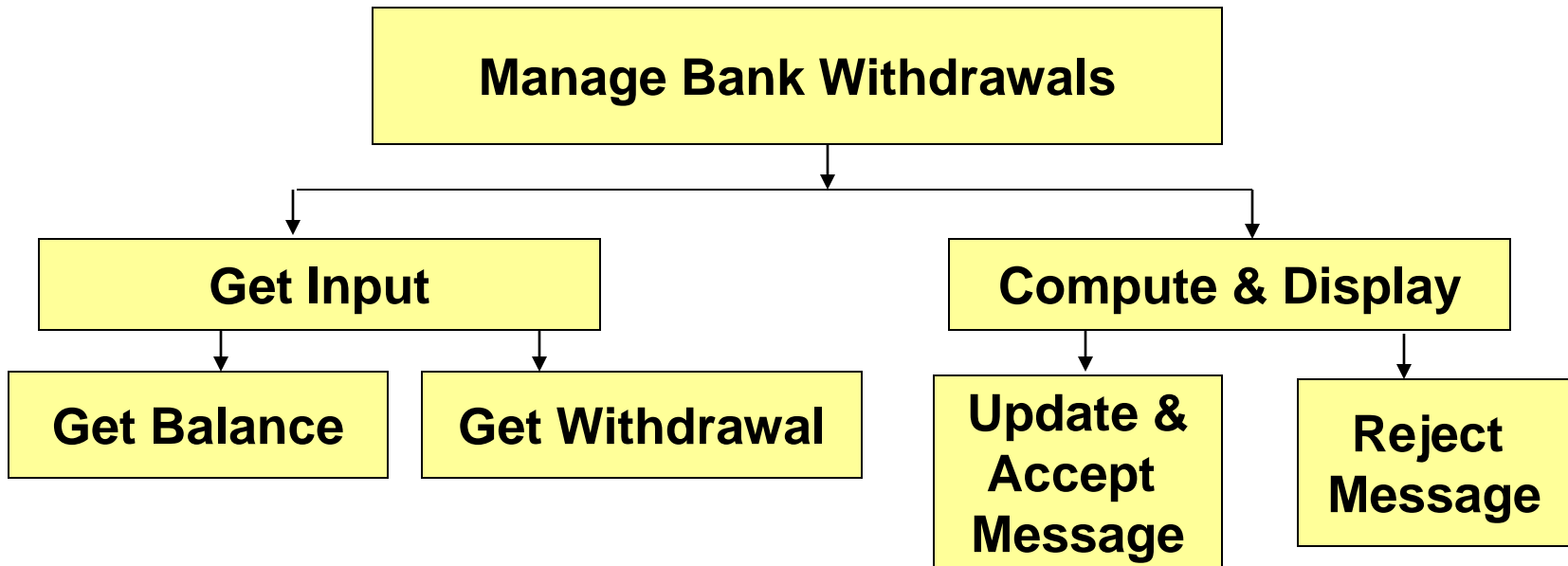
Enter withdrawal: 125

Withdrawal rejected.

New balance: 50

Program Design

- Many novice programmers start coding all the details (formatting) before working out the main components
- This hampers problem solving
- Using Top-Down Design Problems are subdivided into sub-problems which are repeatedly subdivided each unit are of a manageable size.



Initial Design & Refinement

Get balance

display prompt
get balance value

Get withdrawal amount

display prompt
get withdrawal amount

Decide accept or refuse and compute new balance

if balance greater than or equal to withdrawal, accept

display accept message
compute new balance
display new balance

otherwise reject display reject message

display reject message
display new balance

Implementation

```
import java.io.*;
class AccountManagement
{   public static void main (String[] args)
        throws IOException
    {   // Declaring primitives and references
        BufferedReader stdin = new BufferedReader
            (new InputStreamReader (System.in));
        String initString, withdrawString;
        int initialBalance, withdrawal;

        // Getting the balance
        System.out.print ("Enter initial balance");
        initString = stdin.readLine();
        initialBalance = Integer.parseInt (initString);

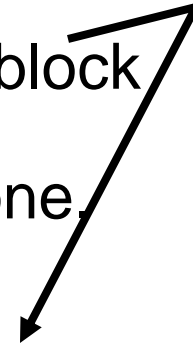
        // Getting the withdrawal amount
        System.out.print("Enter withdrawal");
        withdrawString = stdin.readLine();
        withdrawal = Integer.parseInt (withdrawString);
```

Implementation

```
if (initialBalance >= withdrawal) // sufficient funds
{
    System.out.println("Withdrawal accepted");
    System.out.print("New balance: ");
    System.out.println(initialBalance - withdrawal);
}
else // insufficient funds
{
    System.out.println("Withdrawal rejected");
    System.out.print("New balance: ");
    System.out.println(initialBalance);
}
} // end of main()
} // end of class
```

In the last program we used ...

- the **if ... else** statement
- the “greater than or equal to” operator `>=`
- the use of curly braces `{` and `}` to make a block
 - so several statements are considered as one.



```
if (initialBalance >= withdrawal)
{
    System.out.println("Withdrawal accepted");
    System.out.print("New balance: ");
    System.out.println(initialBalance - withdrawal);
}
```


The **if...else** statement

- main statement for decision making
- Its general form is:

```
if (condition)
    statement      // executed if condition true
else statement    // executed if condition false
```

- or, if using blocks,

```
if (condition)
{
    statements;    // executed if condition true
}
else
{
    statements;    // executed if condition false
}
```

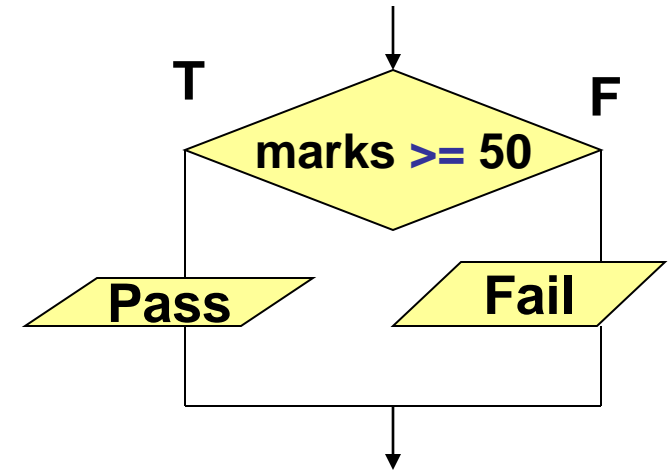
The rules ...

- **condition** must be a **boolean** condition - an expression that evaluates to **true** or **false**.
- If **true** the first statement is executed; otherwise the second statement is executed.
- If there are more than one statement they are included in a block { ... }
- The individual **statements** to be executed are statements, so they include a **';** at the end;
- the **else** is optional. If not present, the syntax is:

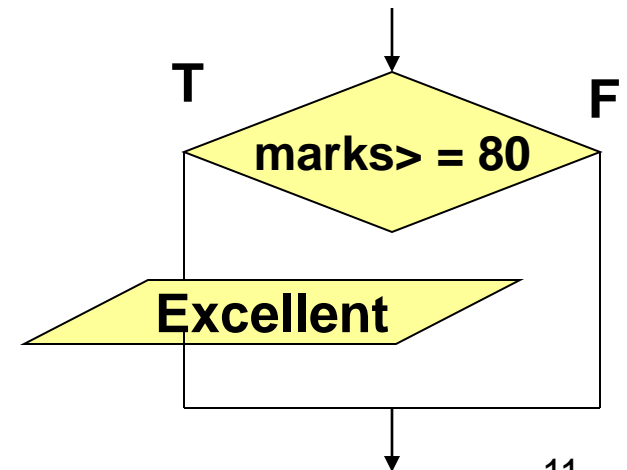
```
if (condition) {  
    statements;  
}
```

Comparing if and if ... else

```
// prints Pass if marks >= 50
// prints Fail otherwise
if (marks >= 50)
    System.out.println("Pass");
else
    System.out.println("Fail");
```



```
// prints Excellent if marks >= 80
if (marks >= 80)
    System.out.println("Excellent");
```



Quiz

The following program segment is attempting to print messages based on Programming 1 final marks. Identify all the errors

```
if  marks >= 50;  
    System.out.println("You passed Prog.1");  
    System.out.println("You may proceed to Prog.2");  
else  
    System.out.println("You failed Prog.1");  
    System.out.println("You have to repeat Prog.2");
```

Exercise

Complete the program below to print whether a number is odd or even in the format below.

Sample Input/output

Enter a positive integer : 37

37 is odd

```
import java.util.*;
public class Selection1
{   public static void main(String args[])
    {   Scanner sc = new Scanner(System.in);
        System.out.print("Enter a positive integer : ");
        int num = sc.nextInt();
        ...
        ...

    }
}
```

Exercise

- (a) Complete program to print welcome message in the form
Welcome Ms.XXXX or Welcome Mr.xxxx depending on gender
- (b) Modify program such that if gender is 'M' or 'm' use title "Mr", 'f' or 'F' use title "Ms" otherwise no title.

```
import java.util.*;
public class TestSelection
{   public static void main(String args[])
    {   Scanner console=new Scanner(System.in);
        String name;
        char gender;
        System.out.print("Enter your name : ");
        name = console.nextLine();
        System.out.print("Enter gender (M/F) :");
        gender = console.nextLine().charAt(0);
        ...

    }
}
```

Exercise : Selection

(Using Relational and Logical Operators)

Complete the program below to print whether year is a leap year or normal year. A leap year is divisible by 4 but not 100 or is divisible by 400.

Sample Input/output

```
Enter year : 1980
1980 is leap year
Enter year : 1900
1900 is normal year
```

```
import java.util.*;
public class Selection2
{   public static void main(String args[])
    {   Scanner sc = new Scanner(System.in);
        System.out.print("Enter year : ");
        int year = sc.nextInt();

    }
}
```

Quiz: String Comparison

```
import java.util.*;
public class TestString
{   public static void main(String args[])
    {
        String s1 = "Abraham";
        String s2 = "Isaac";
        if ( s1.compareTo(s2) > 0 )
            System.out.println(s1 + " is greater");
        else if ( s1.compareTo(s2) == 0)
            System.out.println(s1 + " same as " + s2 );
        else
            System.out.println(s2 + " is greater");
    }
}
```

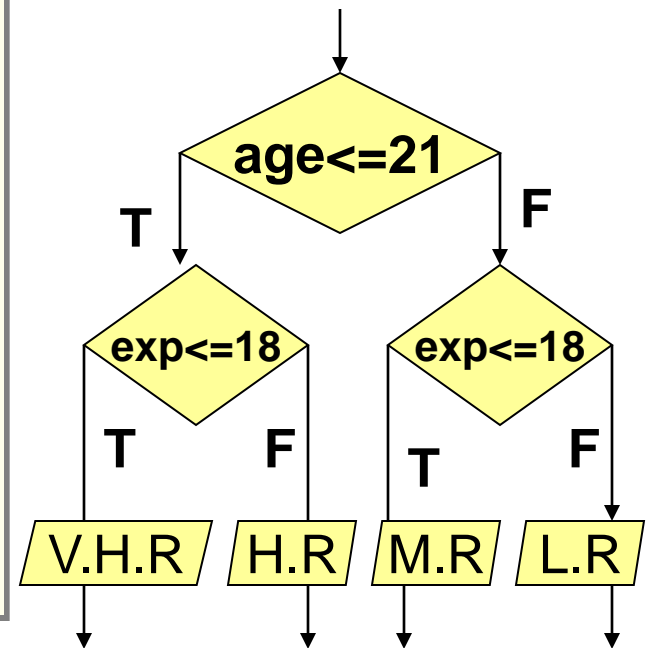

Nested if ... else

Imagine that you have joined a motor insurance company

You are asked to print a chart based on age and experience (months)


```
if (age <= 21)
    if (exp <= 18)
        System.out.print("V.H.R")
    else System.out.print("H.R")
else
    if (exp <= 18)
        System.out.print("M. Risk");
    else System.out.print("L. Risk");
```

	exp <= 18 m	exp > 18 m
age<=21	Very High Risk	High Risk
age>21	Moderate Risk	Low Risk



Nested if ... else

```
if (mark >= 50)    // pass the subject
    if (mark >= 80)
        System.out.println("Well done!");
    else
        System.out.println("Passed the subject");
```



Input

mark = 85

mark = 65

mark = 25

mark = 125

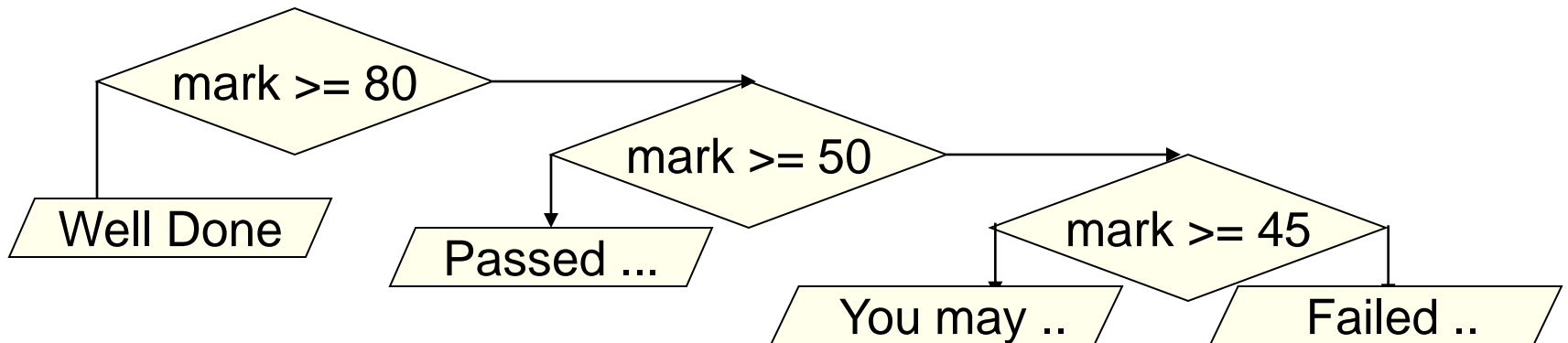
Guess the output

Note that else is associated with the closest if

What changes needed to print Failed the subject if mark < 50 ?

Another way

```
if (mark >= 80)
    System.out.println("Well done!");
else if (mark >= 50)
    System.out.println("Passed the subject");
else if (mark >= 45)
    System.out.println("You may get another chance!");
else
    System.out.println("Failed the subject");
```



Quiz What would you change in the above program to print
Incorrect Marks if mark > 100 ?

Summary of Decision Making

```
if (Boolean_expression)  
    statement;           // simple decision
```

```
if (Boolean_expression)  
    statement;  
else  
    statement;           // else is optional
```

```
if (Boolean_expression1)  
    statement;  
else if (Boolean_expression2)  
    statement;           // 0 or more else if  
else  
    statement;           // else is optional
```

Exercise (Using Nested if)

Complete the program below to read the weight in kg and print Under-Weight if weight is less than 60, Normal-Weight if weight is between 60 and 80 and Over-Weight if weight is over 80.

Sample Input/output

Enter weight : 72.0

Normal-Weight

Enter weight : 95.0

Over-Weight

```
import java.util.*;
public class Selection3
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter weight : ");
        int weight = sc.nextInt();
        ...
        ...
    }
}
```

Exercise (multiple selection criteria)

- Required to write a program to compute car insurance.
- Input value of car, driving experience of insurer (in months) and age of car.
- Insurance consists of basic part + premium.
- Basic part is 1% of car value + \$200.
- Premium based on table below.
- Complete the program on next slide.

Experience	>=30 months	<30months
Age of car		
<= 5 years	\$100	\$250
> 5years	\$150	\$325

Car Insurance Computation (multiple selection)

```
import java.util.*;
public class Selection4
{   public static void main(String args[])
    {   Scanner sc = new Scanner(System.in);
        System.out.print("Enter car value : ");
        double value = sc.nextDouble();
        System.out.print("Enter exp in months : ");
        int exp = sc.nextInt();
        System.out.print("Enter age of car (years) : ");
        int age = sc.nextInt();
        double insurance = value * 0.01;    // 1%

        System.out.println("Insurance = " + insurance);
    }
}
```

Application (Nested if Exercise)

- Write a program to input the salary and compute the tax based on table next.
- Complete the missing parts.

Salary Range	Tax
<20,000	15%
20,000-40,000	20%
>40,000	30

Tax Computation (Using Nested if)

What will the output of the program below?

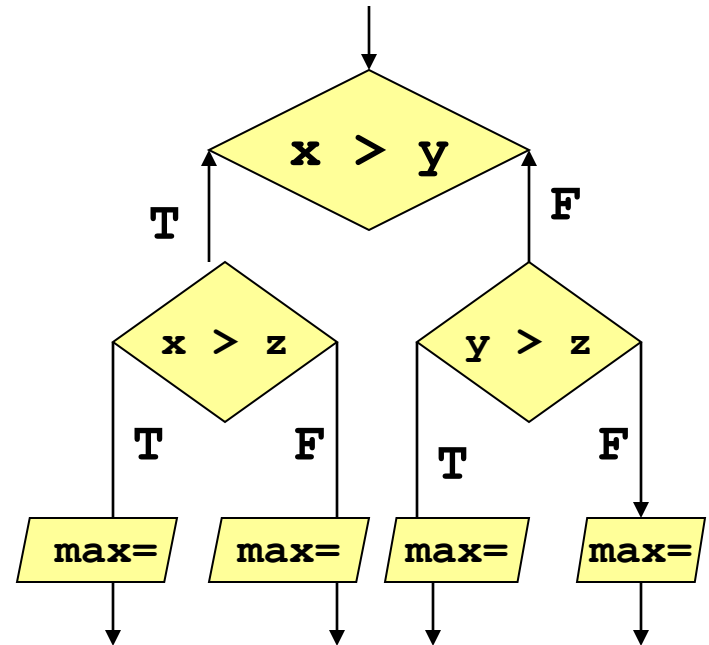
```
import java.util.*;
public class Selection5
{   public static void main(String args[])
    {   Scanner sc = new Scanner(System.in);
        System.out.print("Enter salary : ");
        double salary = sc.nextDouble();
        double tax;

        System.out.println("Tax = " + tax);
    }
}
```

Ans: _____

Quiz - Nested If

```
if( x > y )
    if ( x > z )
        max = _;    // 1
    else
        max = _;    // 2
else
    if ( y > z )
        max = _;    // 3
    else
        max = _;    // 4
```



Fill in the 4 blanks labelled 1 to 4.

Conditional Operator

The conditional operator is used as a shorthand for an **if...else** statement.

```
if (x < y)
    minVal = x;
else
    minVal = y;
```



```
minVal = (x < y) ? x : y;
```

Equivalent
statement using
Conditional
operator

The general case is:

```
condition ? yesExpression : noExpression;
```

The condition is evaluated first; if **true** the value of the entire expression is *yesExpression*, otherwise the value is *noExpression*.

Quiz:What is the output ?

```
int n = 7;  
System.out.println("The number is: "  
    + ((n % 2) == 0 ? "even" : "odd"));
```

The **switch** statement

- Java **switch** statement for multiway decisions, is an alternative to nested **if...else**.
- The following code prints the month name corresponding to a month number:

```
switch (month) {  
    case 1 : System.out.println("January");  
        break;  
    case 2 : System.out.println("February");  
        break;  
    case 3 : System.out.println("March");  
        break;  
    case 4 : System.out.println("April");  
        break;  
    case 5 : System.out.println("May");  
        break;  
}
```

```
case 6 : System.out.println("June");
    break;
case 7 : System.out.println("July");
    break;
case 8 : System.out.println("August");
    break;
case 9 : System.out.println("September");
    break;
case 10 : System.out.println("October");
    break;
case 11 : System.out.println("November");
    break;
case 12 : System.out.println("December");
    break;
default : System.out.println("Not a valid month");
    break;
}
```

Sample Input/output

Enter a month(1-12):

5

May

Rules for **switch** statement

- The selector must be an integer-type expression
- If the selector matches any of the values in the cases, the execution continues there.
- The **break** is needed to avoid the execution of the statements following the match. After the break execution continues with the first statement after the switch.
- If there is no match on any of the cases, control transfers to a **default** case.
- If there is no **default** case the entire **switch** is terminated and execution continues with the first statement after the **switch**.

Handling several alternatives with Switch

```
switch (month) {  
    case 1 :  
    case 2 :  
    case 3 : System.out.println("First Quarter");  
        break;  
    case 4 :  
    case 5 :  
    case 6 : System.out.println("Second Quarter");  
        break;  
    case 7 :  
    case 8 :  
    case 9 : System.out.println("Third Quarter");  
        break;  
    case 10 :  
    case 11 :  
    case 12 : System.out.println("Fourth Quarter");  
        break;  
    default : System.out.println("Wrong Input!");  
        break;  
}
```

**months 1,2 3
fall through to
this
statement.**

